	Application No.	Applicant(s)
Notice of Allowability	09/693,517	CROWL ET AL.
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--
All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *2/21/2006*.

2. ☒ The allowed claim(s) is/are *1,3,5-8,10-18, and 20 (now renum 1-16)*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some*   c) ☐ None   of the:

        1. ☐ Certified copies of the priority documents have been received.

        2. ☐ Certified copies of the priority documents have been received in Application No. _____ .

        3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

    * Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.

    (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached

        1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____ .

    (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of

        Paper No./Mail Date _____ .

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)
1. ☐ Notice of References Cited (PTO-892)

2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)

3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____

4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material

5. ☐ Notice of Informal Patent Application (PTO-152)

6. ☒ Interview Summary (PTO-413),
    Paper No./Mail Date *4/28/06* .

7. ☒ Examiner's Amendment/Comment

8. ☒ Examiner's Statement of Reasons for Allowance

9. ☐ Other _____ .

DETAILED ACTION

1.   This action is responsive to the Applicant's response filed 2/21/2006.

As indicated in Applicant's response, claims 1, 10, 14, 16 and 17 have been amended.

Claims 1, 3-8, 10-20 are pending in the office action.

EXAMINER'S AMENDMENT

2.   An examiner's amendment to the record appears below. Should the changes and/or

additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR

1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the

payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with

Fred Mollborn, Reg # 48587 on 4/27/06.

The application has been amended as follows.

In the CLAIMS:

Claim 1:

A method of compilation of a source program written in one of the C++ and Ada

programming languages into an object file, using one or more ~~associated~~ libraries of instances,

wherein each instance is a specialization of a generic template and its available operations <u>and is</u>

<u>identified by a linker symbol name, and wherein the generic template is expressed in the same</u>

<u>programming language as the source program,</u> the method comprising<u>, during compilation of the</u>

<u>source program</u>:

~~identifying one or more instances available for use in the one or more libraries using~~

~~linker symbol names for the one or more instances;~~

examining linker symbol names in linker tables and instance names made available in the one or more libraries;

selecting linker symbol names that are likely to correspond to instances available for use in the one or more libraries;

saving the selected linker symbol names in an instance name storage;

receiving a ~~first~~ request to create ~~a first~~ an instance ~~during compilation of the source program~~;

examining the instance name storage to determine ~~determining~~ whether the ~~first~~ instance has been identified in the one or more libraries; ~~and~~

when the ~~first~~ instance has not been identified in the one or more libraries, creating the ~~first~~ instance, and

when the ~~first~~ instance has been identified in the one or more libraries, using the linker symbol name of the ~~first~~ instance as a reference to the ~~first~~ instance already contained within the one or more libraries; and

using either the newly created instance or the identified instance already contained within the one or more libraries to create the object file. ~~, thereby avoiding duplication of instances already contained within the one and more libraries and reducing the time and amount of work needed for compiling the source program.~~

Claim 4:   (Canceled)

Claim 5:

A method as recited in claim [[4]] 1, wherein the examining of symbol tables is done to extract all linker symbol names that are likely to correspond to instances.

Claim 6:

A method as recited in claim [[4]] 1, wherein the selecting of the linker symbol names that are likely to correspond to instances is done by selecting linker symbol names that include a predetermined sequence of characters.

Claim 7:

A method as recited in claim [[4]] 1, wherein the saving of the selected linker symbol names is done by using a hash table.

Claim 8:

A method as recited in claim [[4]] 1,

wherein determining whether the first instance has been identified in the one or more libraries further comprises:

obtaining a first linker symbol name for the first instance;

comparing the first linker symbol name with those selected linker symbol names in the instance name storage that are likely to correspond to template instances, and

wherein creating the first instance operates to create the first instance when the first linker symbol does not match any of those selected linker symbol names that are likely to correspond to template instances.

Claim 10:

A compiler system, embodied in a computer readable medium, the compiler system being

operable to compile source programs written in one of the C++ and Ada programming languages

into object files, the compiler system comprising:

a source program;

a library including at least one instance available for use by the source program, the at

least one instance being a specialization of a generic template and its available operations and

being identifiable by a linker symbol name, the generic template being expressed in the same

programming language as the source program; and

an enhanced compiler operable to compile source code into object files, wherein the

enhanced compiler is operable to:

access the library to identify the at least one instance available in the library by the linker

symbol name of the at least one instance in a linker table and instance names made available in

the library, the enhanced compiler thereby avoiding duplication of instances already contained

within the one and more libraries and reducing the time and amount of work needed for

compiling the source programs

use the linker symbol name of the instance as a reference to the instance already

contained in the library, and

use the identified instance within the library in creating the object files.

Claim 11:

A compiler system as recited in claim 10, wherein the enhanced compiler further

comprises:

an instance <u>name</u> extractor for extracting <u>from the library</u> the at least one instance

available for use by the source program.

Claim 14:

A method of compilation of a source program written in one of the C++ and Ada

programming languages into an object file, using one or more associated libraries ~~with~~ <u>of</u>

instances ~~available for use by the source program~~, wherein each instance is a specialization of a

generic template and its available operations <u>and is identified by a linker symbol name, and</u>

<u>wherein the generic template is expressed in the same programming language as the source</u>

<u>program,</u> the method comprising <u>during compilation of the source program</u>:

examining a linker ~~name~~ table <u>and instance names made available in</u> ~~of~~ the one or more

associated libraries;

extracting from the linker name table one or more linker symbol names that are likely to

correspond to instances;

storing the one or more linker symbol names that have been extracted as one or more

stored linker symbol names;

receiving a ~~first~~ request to create ~~a first~~ <u>an</u> instance ~~during compilation of the source~~

~~program~~, said ~~first~~ instance having a ~~first~~ linker symbol name;

comparing the ~~first~~ linker symbol name with the one or more stored linker symbol names;

~~and~~

creating the ~~first~~ instance only when said comparing indicates that the ~~first~~ linker

symbol name is not one of the stored linker symbol names~~, thereby avoiding duplication of~~

~~instances already contained within the one and more libraries and reducing the time and amount~~

~~of work needed for compiling the source program;~~ and

using the stored linker symbol name corresponding to the linker symbol name in creating

the object file when said comparing indicates that the linker symbol name is one of the stored

linker symbol names.

Claim 17:

A computer readable medium including computer program code for compilation of a

source program written in one of the C++ and Ada programming languages into an object file,

using one or more ~~associated~~ libraries of instances ~~having instances available for use by the~~

~~source program,~~ wherein each instance is a specialization of a generic template and its available

operations and is identified by a linker symbol name, and wherein the generic template is

expressed in the same programming language as the source program, the computer readable

medium comprising instructions operable to execute during compilation of the source program

to:

~~computer program code for identifying one or more instances available for use in the one~~

~~or more libraries, using linker symbol names for the one or more instances;~~

examine linker symbol names in linker tables and instance names made available in the

one or more libraries;

select linker symbol names that are likely to correspond to instances available for use in

the one or more libraries;

save the selected linker symbol names in an instance name storage;

~~computer program code for receiving~~ <u>receive</u> a ~~first~~ request to create ~~a first~~ <u>an</u> instance ~~during compilation of the source program~~;

~~computer program code for determining~~ <u>examine the instance name storage to determine</u> whether the first instance is available for use in the one or more libraries; ~~and~~

~~computer program code for~~

when the ~~first~~ instance has not been identified in the one or more libraries, ~~creating~~ <u>create</u> the ~~first~~ instance, and

when the ~~first~~ instance has been identified in the one or more libraries, ~~using~~ <u>use</u> the linker symbol name of the ~~first~~ instance as a reference to the ~~first~~ instance already contained within the one or more libraries~~, thereby avoiding duplication of instances already contained within the one and more libraries and reducing the time and amount of work needed for compiling the source program~~; <u>and</u>

<u>use either the newly created instance or the identified instance already contained within the one or more libraries to create the object file.</u>

Claim 18:

A computer readable medium as recited in claim 17, wherein the computer program code for creating the ~~first~~ instance ~~operates~~ <u>includes instructions operable</u> to create the ~~first~~ instance when the linker symbol name for the first instance does not match any of the identified linker symbol names for instances available for use in the one or more libraries.

Claim 19:   (Canceled)

Claim 20:

A computer readable medium as recited in claim [[19]] 17, wherein the ~~computer~~

~~program code selecting of~~ instructions operable to select the linker symbol names that are likely

to correspond to instances ~~is done by~~ include instructions to selecting linker symbol names that

include a predetermined sequence of characters.

EXAMINER'S STATEMENT OF REASONS FOR ALLOWANCE

3.   Claims 1, 3, 5-8, 10-18, and 20 are allowed.

The following is an examiner's statement of reasons for allowance.

The prior art taken separately or jointly does not suggest or teach the following features.

A method or computer product for compiling a source program source written in C++

and Ada into a object file, comprising (i) using one or more libraries having instances for use by

the program, wherein each instance is a specialization of a template including its operations, such

template expressed in the same program language as the source program and identified by a

linker symbol name, compiling by means of (ii) examining and selecting instance names from a

linker table that are likely to correspond to instance names made available in the one or more

libraries; and storing the selected instance names in a instance name store; and (iii) upon

receiving a request to create an instance, either creating the instance if it is determined that the

stored instance name has not been identified in the one or more libraries; or use the symbol

linker name as reference to the instance if the instance is determined to already exist within the

one or more libraries, and use such instance name to create the object file; all of which steps

being recited in claims 1, 10, 14, and 17.

Burch, USPN: 6,308,320, discloses reducing at linking time the number of compilation

with use of a depository of object files; thereby avoiding recreation of object files by checking if

an object file already exists in the depository; but Burch fails to suggest or teach the instance as

being a specialized templates written in the same program source like C++/Ada, stored with its

available operations in libraries of instances and accessible each a via linker symbol name as in

(i), such that the instances can be selected for likely use during compilation as recited in (ii); and

the compilation process of creating instances which is based upon determining if any of these

instances has been already made available in such libraries as a result of examining linker table

names and selecting of instances in step (ii)

Any comments considered necessary by applicant must be submitted no later than the

payment of the issue fee and, to avoid processing delays, should preferably accompany the issue

fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for

Allowance."

Conclusion

4. Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The

examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

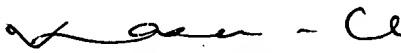supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is

assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before

using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-

272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
April 28, 2006

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100